



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,622	11/24/2003	David B. Lomet	MSFT-2756/302351	4249
41505	7590	09/06/2007 WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION) CIRA CENTRE, 12TH FLOOR 2929 ARCH STREET PHILADELPHIA, PA 19104-2891		
		EXAMINER LE, MIRANDA		
		ART UNIT	PAPER NUMBER	2167
		MAIL DATE	DELIVERY MODE	09/06/2007 PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/720,622	LOMET ET AL.
	Examiner	Art Unit
	Miranda Le	2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 15 June 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-25 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-25 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____. | 6) <input type="checkbox"/> Other: _____. |

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 06/15/2007 has been entered.

2. This communication is responsive to Amendment, filed 06/15/2007.

Claims 1-25 are pending in this application. Claims 1, 15, 25 are independent claims. In the Amendment, claims 1, 15, 25 have been amended. This action is made non-Final.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1, 3, 5-7, 9-14, 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet et al. (US Patent No. 6,182,086), in view of Cotner et al. (US Patent No. 5,884,327).

As per claim 1, Lomet teaches a system for optimizing recovery logging, the system comprising:

a storage device (*i.e. a stable log, col. 4, lines 54-61*) on a calling component machine (*i.e. clients, col. 4, lines 54-61*) (*i.e. Both the server and the client have a volatile main memory with a log buffer, a non-volatile memory with a stable log, and a processing unit, col. 4, lines 54-64*);

a calling component module on the calling component machine, the calling component module adapted to sending a first message (*i.e. During a common interaction, the client-side application sends a request for services (e.g., read a file, return some information, process data, etc.) to the server, col. 4, line 65 to col. 5, line 8*) without logging the first message to the storage device (*i.e. avoid their logging so as to minimize the client's logging cost, col. 11, line 55 to col. 12, line 2*), and wherein the calling component module is adapted to sending a second message to the called component (*i.e. The client sends a stability notification, col. 5, lines 55-67*), and logging a return message (*i.e. The reply messages from the server are kept in the message lookup table at the client. The log records for external input are forced to the stable log file 118 immediately, col. 11, line 55 to col. 12, line 2*) to the storage device when the second message to the called component is sent, wherein logging of the return message is an only forced logging event (*i.e. The log records for external input are forced to the stable log file 118 immediately. None of the other messages needs to be forced to the stable log file, col. 11, lines 55 to col. 12, line 2*) on the calling component module enabling optimized recovery of incomplete requests in

Art Unit: 2167

event of a system crash based on logged messages on the calling component module and the called component (*i.e. If the client subsequently crashes, the client will attempt to restart the client-side application using its stable log file during recovery (col. 5, lines 18-28); and a called component table (*i.e. server*) on the calling component machine for storing information (*i.e. The server and client use the MSNs to identify and track individual messages exchanged between them, col. 10, lines 24-33) associated with the return message (*i.e. The server 54 generates a log record for each of its own write operations on database objects and each reply message, col. 10, lines 35-41).***

Lomet does not specifically teach a called component of a plurality of called components receiving calls from the calling component module, wherein a contract between the calling component module and the called component requires the called component to guarantee persistence of its last return message to the calling component module until released by receiving a second message from the calling component module, wherein the calling component sends the first message.

Cotner teaches a called component of a plurality of called components receiving calls from the calling component module, wherein a contract between the calling component module and the called component requires the called component to guarantee persistence of its last return message to the calling component module until released by receiving a second message from the calling component module, wherein the calling component sends the first message (*i.e. See Fig. 6, (7a) The coordinator sends each of the participants, except the resync server, the committed message, step 601 (FIG. 6). The participant force-writes a commit log record, sends an explicit or implied forget message to the coordinator, commits the unit of work, and forgets it, step 602*

(FIG. 6). In a preferred embodiment, the coordinator can specify whether the coordinator wants an implied or explicit forget from the participant. If an implicit forget is desired, then the next SQL request from the participant implies the forget message (7b) The coordinator sends each of the participants, except the resync server, the rollback message, step 603. The participant writes a non-forced rollback log record, rollbacks the unit of work, and forgets it, step 604. (8a) After receiving the forget responses (either explicitly or implicitly) from all the participants, the coordinator sends a forget message to the resync server, step 605 (FIG. 6). (9a) The resync server writes a non-forced forget log record, and forgets the unit of work, step 606. The resync server is no longer in the "committing" state, col. 10, lines 16-35).

It would have been obvious to one of ordinary skill of the art having the teaching of Lomet and Cotner at the time the invention was made to modify the system of Lomet to include the limitations as taught by Cotner.

One of ordinary skill in the art would be motivated to make this combination in order to have the coordinator sent each of the participants the rollback message in view of Cotner, as doing so would give the added benefit of eliminating the exposure of a local log failure of a client/coordinator while still allowing database resources to be protected by a two-phase commit as taught by Cotner (Summary).

As per claim 25, Lomet teaches a computer-readable storage medium including computer-executable instructions for:

sending a first call message to a called component (*i.e. server, col. 4, lines 54-61*) from a calling component (*i.e. clients, col. 4, lines 54-61*);

sending a second call message to the called component from the calling component (*i.e.*

The client sends a stability notification, col. 5, lines 55-67;

logging a return message to the first call message in a stable log on the called component machine (*i.e. server*) (*i.e. the server's resource manager records the reply in the log buffer and commits the reply record to the stable log prior to sending the reply back to the client (col. 5, lines 8-17; The server 54 forces a reply log record to the stable log 96, by flushing the database log buffer 88, before the reply message is sent to the client 52, col. 10, lines 41-50).*

logging the return message (*i.e. The reply messages from the server are kept in the message lookup table at the client. The log records for external input are forced to the stable log file 118 immediately, col. 11, line 55 to col. 12, line 2*) to the first call message in a stable log on the calling component machine (*i.e. client*) when the second call message to the called component is sent (*i.e. The client sends a stability notification, col. 5, lines 55-67*).

Lomet does not teach wherein a contract between the calling component and the called component requires the called component to guarantee persistence of its last return message to the calling component until released by receiving a second message from the calling component.

Cotner teaches a contract between the calling component and the called component requires the called component to guarantee persistence of its last return message to the calling component until released by receiving a second message from the calling component (*i.e. See Fig. 6, (7a) The coordinator sends each of the participants, except the resync server, the committed message, step 601 (FIG. 6). The participant force-writes a commit log record, sends an explicit or implied forget message to the coordinator, commits the unit of work, and forgets it, step 602 (FIG. 6). In a preferred embodiment, the coordinator can specify whether the*

Art Unit: 2167

coordinator wants an implied or explicit forget from the participant. If an implicit forget is desired, then the next SQL request from the participant implies the forget message (7b) The coordinator sends each of the participants, except the resync server, the rollback message, step 603. The participant writes a non-forced rollback log record, rollbacks the unit of work, and forgets it, step 604. (8a) After receiving the forget responses (either explicitly or implicitly) from all the participants, the coordinator sends a forget message to the resync server, step 605 (FIG. 6). (9a) The resync server writes a non-forced forget log record, and forgets the unit of work, step 606. The resync server is no longer in the "committing" state, col. 10, lines 16-35).

It would have been obvious to one of ordinary skill of the art having the teaching of Lomet and Cotner at the time the invention was made to modify the system of Lomet to include the limitations as taught by Cotner.

One of ordinary skill in the art would be motivated to make this combination in order to have the coordinator sends each of the participants the rollback message in view of Cotner, as doing so would give the added benefit of to eliminate the exposure of a local log failure of a client/coordinator while still allowing database resources to be protected by a two-phase commit as taught by Cotner (Summary).

As per claim 3, Lomet teaches the system of claim 1, wherein the called component table comprises a log sequence number (*i.e. The server and client use the MSNs to identify and track individual messages exchanged between them, col. 10, lines 24-33*) associated with the called component for determining a status of the return message (*i.e. The server 54 generates a log*

record for each of its own write operations on database objects and each reply message, col. 10, lines 35-41).

As per claim 5, Lomet teaches the system of claim 1, wherein a status of the called component's return message is reset when the message (*i.e. logs reply messages, col. 7, lines 40-49*) is logged to the storage device (*i.e. the client sends stability notifications to the server to release previously logged reply log records; The server then removes the reply log record from the message lookup table 100, col. 7, lines 40-64*).

As per claim 6, Lomet teaches the system of claim 1, wherein the storage device stores a stable log associated with the calling component (*i.e. During a common interaction, the client-side application sends a request for services (e.g., read a file, return some information, process data, etc.) to the server, col. 4, line 65 to col. 5, line 8*).

As per claim 7, Lomet teaches the system of claim 1, further comprising a storage device for storing a stable log associated with the called component (*i.e. The server 54 generates a log record for each of its own write operations on database objects and each reply message, col. 10, lines 35-41*).

As per claim 9, Lomet teaches the system of claim 1, wherein the calling component (*i.e. client*) persists over a system failure (*i.e. If the client subsequently crashes, the client will attempt to restart the client-side application using its stable log file during recovery, col. 5, lines 18-28*).

As per claim 10, Lomet teaches the system of claim 1, wherein the called component (*i.e. server*) persists over a system failure (*i.e. The server 54 provides request recovery in one of three ways. One approach is to log the request actions at the server in a manner that enables "undo" recovery. With this approach, the log enables the server to "undo" incomplete requests during recovery. If the server knows that the reply to the original request has not yet been sent to the client, the server can relax the obligation of deterministic replay as long as all database writes of concurrently executed requests are kept isolated. The server can then undo a request and re-execute it all over again when the client re-submits the request, as if it were a new request, col. 10, lines 51-58).*

As per claim 11, Lomet teaches the system of claim 1, wherein the stable log associated with the calling component (*i.e. client*) persists over a system failure (*i.e. If the client subsequently crashes, the client will attempt to restart the client-side application using its stable log file during recovery, col. 5, lines 18-28.*

As per claim 12, Lomet teaches the system of claim 1, wherein a stable log associated with the called component (*i.e. server*) persists over a system failure (*i.e. The server 54 provides request recovery in one of three ways. One approach is to log the request actions at the server in a manner that enables "undo" recovery. With this approach, the log enables the server to "undo" incomplete requests during recovery. If the server knows that the reply to the original request has not yet been sent to the client, the server can relax the obligation of deterministic replay as long as all database writes of concurrently executed requests are kept isolated. The server can*

then undo a request and re-execute it all over again when the client re-submits the request, as if it were a new request, col. 10, lines 51-58).

As per claim 13, Lomet teaches the system of claim 1, wherein only the last message sent by the called component is guaranteed to be stable stored with the called component (*i.e. the client sends stability notifications to the server to release previously logged reply log records; The server then removes the reply log record from the message lookup table 100, col. 7, lines 40-64*).

As per claim 14, Lomet teaches the system of claim 7, wherein the stable log comprises a component identifier and a message (*i.e. During a common interaction, the client-side application sends a request for services (e.g., read a file, return some information, process data, etc.) to the server, col. 4, line 65 to col. 5, line 8*).

5. Claims 2, 4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet et al. (US Patent No. 6,182,086), in view of Cotner et al. (US Patent No. 5,884,327), and further in view of Britton et al. (US Patent No. 6,401,136).

As per claim 2, Lomet, Cotner do not specifically teach the system of claim 1, further comprising a highest log sequence table for storing a highest log sequence number written to a memory buffer and a highest log sequence number written to the storage device.

However, Britton teaches a highest log sequence table for storing a highest log sequence number written to a memory buffer and a highest log sequence number written to the storage

device (*i.e. Responsive to determining if a new connection has been established, the highest sequence identifier of a message persistently stored in the destination persistent queue is transmitted from the destination device to the source device, col. 3, line 58 to col. 4, line 12*).

It would have been obvious to one of ordinary skill of the art having the teaching of Lomet, Cotner and Britton at the time the invention was made to modify the system of Lomet, Cotner to include the limitations as taught by Britton.

One of ordinary skill in the art would be motivated to make this combination in order to transmit the highest sequence identifier of a message from the destination device to the source device in view of Britton (*col. 3, line 58 to col. 4, line 12*), as doing so would give the added benefit of allowing data to be more quickly available to applications communicating using the message queues as taught by Britton (*col. 3, lines 21-34*).

As per claim 4, Britton teaches the system of claim 2, wherein the status of a return message is determined by comparing the highest log sequence number written to memory and the highest log sequence number written to the storage device and a log sequence number associated with the called component (*i.e. when a request from a user application of the destination device for a message received from the source device is received, it is determined if a received message and an associated sequence identifier are stored in the destination persistent queue at the destination device have an associated sequence identifier which is not greater than a highest sequence identifier associated with a persistently stored message, col. 4, lines 37-51*).

6. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lomet et al. (US Patent No. 6,182,086), in view of Cotner et al. (US Patent No. 5,884,327), and further in view of Ganesh et al. (US Patent No. 6,684,223).

As per claim 8, Lomet, Cotner do not explicitly teach the system of claim 1, wherein the calling component module sends at least one message to a plurality of called component.

However, Ganesh teaches the calling component modules sends at least one message to a plurality of called component (*i.e. coordinating database system 310 transmits two requests to modify data, one request to participating database system 350 to employee.modify a record A in a table AA (not shown), and another to participating database system 390 to modify a record B in a table BB , col. 9, lines 39-50.*)

It would have been obvious to one of ordinary skill of the art having the teaching of Lomet, Cotner and Ganesh at the time the invention was made to modify the system of Lomet, Cotner to include the limitations as taught by Ganesh.

One of ordinary skill in the art would be motivated to make this combination in order to indicate the last-used-log-sequence-number for each participating database system in a transaction in view of Ganesh (*col. 9, lines 5-13*), as doing so would give the added benefit of enabling a log tracking data to be communicated with a message specifically used to transport updates to log tracking data between database systems as taught by Ganesh (*col. 10, lines 39-47*).

7. Claims 15, 22-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shoaib et al. (US Patent No. 7,152,180), in view of Cotner et al. (US Patent No. 5,884,327).

As per claim 15, Shoaib teaches a method of recovery logging (*i.e. Recovery Management, col. 6, line 56 to col. 7, line 14*) comprising:

determining whether a first return message received from a called component has been stably logged on a called component machine (*See Figs. 2, 3, 4*), wherein the first return message is received in response to a first message sent from a calling component to the called component, wherein the calling component resides on a calling component machine different than the called component machine (*i.e. the recovery manager module 52 will ensure that any outbound messages are received by the remote application once it restarts, col. 6, line 56 to col. 7, line 14*);

in response to determining that the first return message has not been stably logged, logging at least the first message to a stable log on the called component before a second message is sent to the calling component (*i.e. FIG. 2 may have multiple options regarding the direction of message logging, including for outgoing messages only, for incoming messages only, or for both directions. In addition, the client 24 may have options for the timing of message logging operations, such as logging messages before sending an outgoing message, after sending an outgoing message or asynchronously. Likewise, the server 26 may log messages before or after delivering an incoming message to the application or asynchronously, col. 5, lines 22-34*), wherein the logging enables recovery of incomplete requests in event of a system crash (*i.e. The reliable messaging system further includes a reliability subsystem having reliability components, which are capable of configurably logging the message, detecting a plurality of failures, notifying a remote entity interconnected with the configurable reliable messaging system via the network of the plurality of failures, and recovering from the plurality of failures, col. 2, line 57 to col. 3, line 15*).

Shoaib does not expressly teach wherein a contract between a calling component and the called component requires the called component to guarantee persistence of its last return message to the calling component until released by receiving a second message from the calling component.

Cotner teaches a contract between a calling component and the called component requires the called component to guarantee persistence of its last return message to the calling component until released by receiving a second message from the calling component (*i.e. See Fig. 6, (7a) The coordinator sends each of the participants, except the resync server, the committed message, step 601 (FIG. 6). The participant force-writes a commit log record, sends an explicit or implied forget message to the coordinator, commits the unit of work, and forgets it, step 602 (FIG. 6). In a preferred embodiment, the coordinator can specify whether the coordinator wants an implied or explicit forget from the participant. If an implicit forget is desired, then the next SQL request from the participant implies the forget. (7b) The coordinator sends each of the participants, except the resync server, the rollback message, step 603. The participant writes a non-forced rollback log record, rollbacks the unit of work, and forgets it, step 604. (8a) After receiving the forget responses (either explicitly or implicitly) from all the participants, the coordinator sends a forget message to the resync server, step 605 (FIG. 6). (9a) The resync server writes a non-forced forget log record, and forgets the unit of work, step 606. The resync server is no longer in the "committing" state, col. 10, lines 16-35).*

It would have been obvious to one of ordinary skill of the art having the teaching of Shoaib and Cotner at the time the invention was made to modify the system of Shoaib to include the limitations as taught by Cotner.

One of ordinary skill in the art would be motivated to make this combination in order to have the coordinator sent each of the participants the rollback message in view of Cotner, as doing so would give the added benefit of eliminating the exposure of a local log failure of a client/coordinator while still allowing database resources to be protected by a two-phase commit as taught by Cotner (Summary).

As per claim 22, Shoaib teaches the method of claim 15, wherein the called component (*i.e. remote device, Fig. 3*) persists over a system failure (*i.e. failure detection, Fig. 3*).

As per claim 23, Shoaib teaches the method of claim 15, wherein the calling component (*i.e. local device, Fig. 3*) persists over a system failure (*i.e. failure detection, Fig. 3*).

8. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shoaib et al. (US Patent No. 7,152,180), in view of Cotner et al. (US Patent No. 5,884,327), and further in view of Ganesh et al. (US Patent No. 6,684,223).

As per claim 21, Shoaib, Cotner do not teach the method of claim 15, further comprising sending at least one message to a plurality of called components before forcing a log.

However, Ganesh teaches the calling component modules sends at least one message to a plurality of called component (*i.e. coordinating database system 310 transmits two requests to modify data, one request to participating database system 350 to employee.modify a record A in a table AA (not shown), and another to participating database system 390 to modify a record B in a table BB , col. 9, lines 39-50*).

It would have been obvious to one of ordinary skill of the art having the teaching of Shoaib, Cotner and Ganesh at the time the invention was made to modify the system of Shoaib, Cotner to include the limitations as taught by Ganesh.

One of ordinary skill in the art would be motivated to make this combination in order to indicate the last-used-log-sequence-number for each participating database system in a transaction in view of Ganesh (*col. 9, lines 5-13*), as doing so would give the added benefit of enabling a log tracking data to be communicated with a message specifically used to transport updates to log tracking data between database systems as taught by Ganesh (*col. 10, lines 39-47*).

9. Claims 16-20, 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shoaib et al. (US Patent No. 7,152,180), in view of Cotner et al. (US Patent No. 5,884,327), and further in view of Britton et al. (US Patent No. 6,401,136)).

As per claim 16, Shoaib, Cotner do not specifically teach the method of claim 15, wherein determining whether the first return message received from the called component has been stably logged comprises:

in response to determining that an entry for the component is not in a table of information associated with called components, adding an entry for the first message to the table, the entry comprising an identifier for the component, and a log sequence number set to a lowest possible value.

However, Britton teaches in response to determining that an entry for the component is not in a table of information associated with called components, adding an entry for the first message to the table, the entry comprising an identifier for the component, and a log sequence number set to a lowest possible value (*i.e. The requesting user application may then be provided*

a message stored in the destination persistent queue with the smallest sequence identifier if a received message and an associated sequence identifier is stored in the destination persistent queue at the destination device has an associated sequence identifier which is not greater than a highest sequence identifier associated with a persistently stored message, col. 4, lines 37-51).

It would have been obvious to one of ordinary skill of the art having the teaching of Shoaib, Cotner and Britton at the time the invention was made to modify the system of Shoaib, Cotner to include the limitations as taught by Britton.

One of ordinary skill in the art would be motivated to make this combination in order to transmit the highest sequence identifier of a message from the destination device to the source device in view of Britton (*col. 3, line 58 to col. 4, line 12*), as doing so would give the added benefit of allowing data to be more quickly available to applications communicating using the message queues as taught by Britton (*col. 3, lines 21-34*).

As per claim 17, Britton teaches the method of claim 16, further comprising:

in response to determining that the entry for the component is in the table of information associated with called components, comparing a highest stably logged log sequence number with the log sequence number in the table entry (i.e. when a request from a user application of the destination device for a message received from the source device is received, it is determined if a received message and an associated sequence identifier are stored in the destination persistent queue at the destination device have an associated sequence identifier which is not greater than a highest sequence identifier associated with a persistently stored message, col. 4, lines 37-51).

As per claim 18, Britton teaches the method of claim 17, further comprising:

in response to determining that the highest stably logged log sequence number is greater than the log sequence number in the entry, proceeding to call the component without forcing the log (*i.e. the committed sequence number received in the message from the receiving device 20 is persistently stored at the sending device to indicate the highest sequence number which has been committed by the receiving device 20 (block 116). It may then be determined if the received message was a resync message from the receiving device 20 (block 118). If the message was not a resync message then the message was a commit message and control waits for the next event. If the message was a resync message then messages in the send persistent message queue 14 which have a higher sequence number than the committed sequence number received in the resync message are transmitted to the receiving device 20, col. 8, line 60 to col. 9, line 6.*)

As per claim 19, Britton teaches the method of claim 17, further comprising:

in response to determining that the highest stably logged log sequence number is less than the log sequence number in the table entry, forcing the log so that the highest stably logged record has a higher log sequence number than the table entry (*i.e. However, if the extracted sequence number is not less than or equal to the local committed sequence number, then the message and the associated extracted sequence number are persistently stored in the receive persistent message queue 24, col. 10, lines 37-49.*)

As per claim 20, Britton teaches the method of claim 19, further comprising updating the highest stably logged log sequence number to the highest log sequence number written to the

stable log (*i.e. then the largest sequence number of a complete sequence of a message persistently stored in the receive persistent message queue 24 is persistently stored as the local committed sequence number (block 226). A commit message is then sent to the sending device 10 which specifies the local committed sequence number, col. 10, line 37-49*).

As per claim 24, Shoaib, Cotner do not teach the method of claim 15, wherein messages sent to the called component are uniquely identified.

Britton teaches messages sent to the called component are uniquely identified (*i.e. The transmitted message has transmitted with it an associated sequence identifier which identifies the message stored in the source persistent queue. The transmitted message and the associated sequence identifier are received at the destination device and the received message and the associated sequence identifier stored in a destination persistent queue at the destination device, col. 2, line 66 to col. 3, line 20*).

It would have been obvious to one of ordinary skill of the art having the teaching of Shoaib, Cotner and Britton at the time the invention was made to modify the system of Shoaib, Cotner to include messages sent to the called component are uniquely identified as taught by Britton.

One of ordinary skill in the art would be motivated to make this combination in order to transmit the highest sequence identifier of a message from the destination device to the source device in view of Britton (*col. 3, line 58 to col. 4, line 12*), as doing so would give the added benefit of allowing data to be more quickly available to applications communicating using the message queues as taught by Britton (*col. 3, lines 21-34*).

Response to Arguments

10. Applicant's arguments regarding the claims, as amended, define the prior arts, with respect to claims 1, 15, 25 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham, can be reached on (571) 272-7079. The fax number to this Art Unit is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Miranda Le
August 27, 2007